RECEIVED
CENTRAL FAX CENTER
NOV 16 2006

REMARKS

Claims 1-6, 8-16, and 18-24 are all the claims pending in the application. Claims 1-22 stand rejected on prior art grounds. Claims 1-19 and 21-22 stand rejected upon informalities. Claims 7 and 17 are cancelled herein without prejudice or disclaimer. Claims 23 and 24 are added. Claims 1, 5, 9, 11, 15, 19, 21, and 22 are amended herein. Applicants respectfully traverse these rejections based on the following discussion.

I. The Objections to the Drawings

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign mentioned in the description: "70". Accordingly, the Applicants have amended the specification to remove the inadvertent inclusion of the reference number "70" and the associated description thereof. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw this objection.

II. The 35 U.S.C. §112, First Paragraph, Rejection

Claims 1-19 and 21-22 stand rejected under 35 U.S.C. §112, first paragraph. These rejections are traversed as explained below. The Office Action concludes that the claimed "building a list of a sequence of said static and dynamic structures" and "building multiple repeating dynamic structures at runtime" are not adequately described in the specification. However, the rejections given in the Office Action are *prima facie* defective according to the Federal Circuit in Fiers v. Sugano, 984 F.2d 1164, 25 USPQ 2d 1601, 1607 (Fed. Cir. 1993)

PAGE 10/31 * RCVD AT 11/16/2006 3:13:53 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/4 * DNIS:2738300 * CSID:3012618825 * DURATION (mm-ss):08-10

(quoting *In re* Marzocchi, 439 F.2d 220, 223, 169 USPQ 267, 269 (C.C.P.A. 1971); Weil v. Fritz, 601 F.2d 551, 555, 202 USPQ 447, 450 (C.C.P.A. 1979)):

> [A] specification disclosure which contains a teaching of the manner and process of making and using the invention in terms which correspond in scope to those used in describing and defining the subject matter sought to be patented *must* be taken as in compliance with the enabling requirement of the first paragraph of §112 *unless* there is reason to doubt the objective truth of the statements contained therein which must be relied on for enabling support...." "[A]ny party making the assertion that a U.S. Patent specification or claims fails, for one reason or another, to comply with §112 bears the burden of persuasion in showing said lack of compliance.

In this regard the Office Action fails to provide a reason to doubt the objective truth of the statements in the specification regarding "building a list of a sequence of said static and dynamic structures" and "building multiple repeating dynamic structures at runtime", and as such, the rejections are *prima facie* defective.

The CCPA has further indicated in *In re* Angstadt, 537 F.2d 489, 190 USPQ 214, 219 (C.C.P.A. 1976) (citing *In re* Armbruster, 512 F.2d 676, 185 USPQ 152 (C.C.P.A. 1975)):

> [T]he PTO has the burden of giving reasons, supported by the record as a whole, why the specification is not enabling.... Showing that the disclosure entails undue experimentation is part of the PTO's initial burden....

In this regard, the Office Action fails to provide a reason why undue experimentation would be required with respect to "building a list of a sequence of said static and dynamic structures" and "building multiple repeating dynamic structures at runtime", and as such, the rejections are *prima facie* defective.

Whether the specification is enabling is judged with respect to whether one of ordinary skill in the art would properly understand how to make and use the invention as it is described in the specification and drawings. The determination of the level of ordinary skill has been articulated by the Federal Circuit in Custom Accessories Inc. v. Jeffrey-Allan Indus., 807 F.2d 955, 1 USPQ 2d 1196, 1201 (Fed. Cir. 1986):

> The person of ordinary skill is a hypothetical person who is presumed to be aware of all the pertinent prior art. The actual inventor's skill is not determinative. Factors that may be considered in determining level of skill include: type of problems encountered in art; prior art solutions to those problems; rapidity with which innovations are made; sophistication of the technology; and educational level of active workers in the field. Not all such factors may be present in every case.

With respect to the above, pages 1-2 of the Applicants' specification, as originally filed, indicates what problems are encountered in the prior art as well as what some of the prior art solutions are with respect to the encountered problems:

> A typical eXtensible Markup Language (XML)-based transaction in a business-to-business (B2B) environment involves combining transaction information such as name, address, social security number, credit card number, etc. from various data sources. Some of this information is fixed for a given trading partner, transaction, and set of business rules. Existing solutions store the trading partner rules and transaction information in a database or file or in an XML format, which is then read and translated to the output XML format in the computer system running the B2B exchange. Unfortunately, in a heavy B2B transaction environment, the incremental costs associated with reading and translating the stored information can be significantly high.
>
> XML files that carry B2B messages have varied static and dynamic content dependent on the trading partner profile (TPP). Within a given business message in an XML format, different business partners require different views of data as defined by the TPP. The result is an XML file that has static sections that are structurally the same but with different views of data. However,

> building a static structure has several disadvantages including a redundant/repetitive code, and that the static structure and content are intertwined with the business logic. Other disadvantages are that the static structure is not modular, which leads to limited reusability, and is not flexible, whereby changes might involve logic from several TPPs. Additionally, static structures are not scalable, wherein the introduction of a new TPP in the electronic B2B exchange requires additional code to be read, entered, and stored. Also, "building" an XML file for a given transaction is slower and inefficient because of building the aforementioned static portion and because of runtime inefficiencies.
>
> Conventional techniques relating to different areas of XML technology exist. For example, XML techniques have been previously described for (1) XML construction - in U.S. Patent 6,635,089 issued to Burkett et al; (2) XML storage - in U.S. Patent 6,643,633 issued to Chau et al.; (3) XML data integrity - in U.S. Patent 6,687,848 issued to Najmi; and (4) XML translation - in U.S. Patent 6,601,071 issued to Bowker et al., the complete disclosures of which in their entireties, are herein incorporated by reference. However, the conventional techniques may not fully provide for a solution that provides for XML file construction, structure integrity, and mass customization, which are three significant areas requiring a solution as identified by the industry.

Furthermore, the technological environment of the field of the Applicants' invention (i.e., computer-implemented systems and methods for improving coding processing in a business-to-business environment) at the time of the invention, was fast-paced with new innovations being developed with great rapidity. This is evidenced by the relative closeness in conception (evidenced by the relative closeness in the respective filing dates) of the four prior art patents discussed in the Applicants' specification (as provided above) as well as the several prior art patents listed in those patents as being relevant. Next, the actual level of sophistication of the technology related to coding processes in B2B environment at the time of the Applicants' invention (2004) was such that several coding techniques and off-the-shelf products were available to facilitate the practical use of the Applicants' claimed invention. For example, page 9

of the Applicants' specification, as originally filed, provides, "Generally, custom generated code based on XML, Java-based XML parsers and Java-based data access tools, and XML editors may be used to implement the method described above." Moreover, page 13 of the Applicants' specification, as originally filed, provides, "Additionally, the invention utilizes tools and applications such as xDoc XML Converter, available from CambridgeDocs, Cambridge, USA, to perform the transformation continuously as needed rather than at install time and tying it with the business partner criteria." Finally, the educational level of active workers in the field are college educated (bachelors and/or masters degrees) engineers and computer scientists having computer programming skills.

In view of the above factors, at the time of the Applicants' invention (2004), the number of those individuals of ordinary skill in the art, both in the United States and abroad, was sufficient and the skill of such individuals to understand the teachings of the Applicants' invention as described in the specification and drawings was at an appropriate level to allow such individuals to make and use the Applicants' invention without undue experimentation. To further prove this assertion, the Applicants offer the following description of what one of ordinary skill in the art would understand within the context of the Applicants' invention and detailed specification/drawings.

An XML-based transaction involves a combination of static data and dynamic data (at transaction execution time) for a given trading partner, transaction and relevant set of business rules. Also the static and dynamic data have to be combined in a certain sequence to derive a complete XML transaction for that trading partner, to preserve the structural integrity of the XML as per the corresponding document type definition (DTD) format. Moreover a set of

related static data in a transaction is typically grouped into a XML sub-structure. Similarly a set of dynamic data can be grouped into a XML sub-structure. A complete XML structure that represents a real business transaction typically comprises of one or more of such static sub-structures and one or more occurrences of such dynamic sub-structures. Furthermore the static and dynamic components should be linked together in correct sequence to derive a complete XML transaction document. This sequence list must be defined for a trading partner and transaction before the said XML transaction can be exchanged with a given trading partner. Also, the complete XML structure thus derived by linking static and dynamic sub-structures in the correct sequence should be validated against the relevant DTD to validate the integrity of the document. An example of building such a sequence of XML sub-structures is shown below, which is something that one of ordinary skill in the art (programmer) would be able to sufficiently derive given the teachings in the Applicants' specification and drawings, as originally filed.

The following example shows pre-built static and dynamic structures for a shipment transaction. A real-life Purchase Order transaction may contain more elements and sub-structures than depicted here:

Pre-built Static XML sub-structure 1:

```
<fromPartner>
        <Name>IBM</Name>
        <DUNS>123456789</DUNS>
</fromPartner>
<toPartner>
        <Name>PartnerA</Name>
        <DUNS>987654321</DUNS>
</toPartner>
```

Pre-built Dynamic XML sub-structure 1 (single occurrence with empty tags):

```
<Shipment>
        <DocumentID> </ DocumentID>
        <DocumentDate> </ DocumentDate>
        <ShipmentType> </ShipmentType>
        <NumberOfItems> </ NumberOfItems>
</Shipment >
```

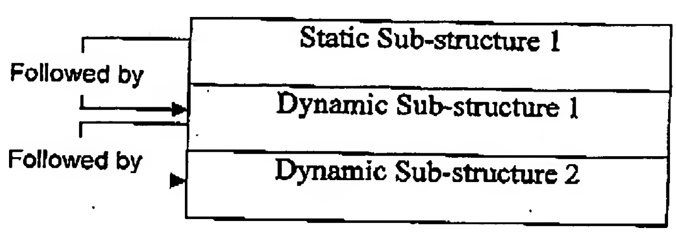
Pre-built Dynamic XML sub-structure 2 (single occurrence with empty tags):

```
<ShipmentItems>
        <PartNumber> </PartNumber>
        <PartDescription> </PartDescription>
        <ItemNumber> </ItemNumber>
        <ItemQuantity> </ItemQuantity>
        <ExpectedDeliveryDate> </ ExpectedDeliveryDate>
</ShipmentItems>
```

Pre-built Dynamic XML sub-structure 3 (single occurrence with empty tags):

```
<ContainerInfo>
        <ContainerId> </ContainerId>
        <ContainerLevel> </ ContainerLevel>
        <ContainerSize> </ ContainerSize>
</ContainerInfo>
```

The following shows a list of sequence of static and dynamic sub-structures associated with trading partner called PartnerA for transaction Shipment:



The complete Shipment XML transaction, based on the above list of sequence, will be constructed at transaction execution time as follows:

```
<ShipmentTransaction>
    <fromPartner>
        <Name>IBM</Name>
        <DUNS>123456789</DUNS>
    </fromPartner>
    <toPartner>
        <Name>PartnerA</Name>
        <DUNS>987654321</DUNS>
    </toPartner>
    <Shipment>
        <DocumentID>SHIP-01</ DocumentID>
        <DocumentDate>2006-01-01</ DocumentDate>
        <ShipmentType>Overnight</ShipmentType>
        <NumberOfItems>1</ NumberOfItems>
    </Shipment>
    <ShipmentItems>
        <PartNumber>PN 01/01</PartNumber>
        <PartDescription>Sample Part</PartDescription>
        <ItemNumber>0001</ItemNumber>
        <ItemQuantity>1000</ItemQuantity>
        <ExpectedDeliveryDate>2006-01-02</ ExpectedDeliveryDate>
    </ShipmentItems>
</ShipmentTransaction>
```

Empty tags filled with actual business values at runtime

The complete XML transaction thus built will be validated against a relevant Document Type Definition (DTD) to validate the structural integrity.

Furthermore, one of ordinary skill in the art would readily understand that, given the Applicants' specification and drawings, for a specified XML transaction linked to a TPP, the transaction-specific business data values are retrieved from various data sources such as database and files at execution time. The actual business data values filled in the dynamic sections of the XML at execution time can vary from one transaction to another, depending on the transaction-context. Hence, the dynamic sub-structures of the XML transaction can occur once or multiple

times depending on the actual business data values retrieved for that transaction at execution time.

One example can be the shipment transaction, where parts are shipped from a vendor to customer, wherein one customer may be shipped one part in a given shipment transaction while another customer may be shipped multiple parts. In a simple case, each of the part detail can be represented in one dynamic sub-structure. The Applicants' invention provides for building dynamic sections with either one instance or multiple instances based on the actual transaction values thus retrieved. An example demonstrating multiple occurrences of dynamic structures is described below, which would be readily understood by one of ordinary skill in the art.

This example is based on Shipment transaction example from above. At transaction execution time (runtime), this example assumes that trading partner called PartnerA is sent a Shipment transaction with 2 items. The computer program as implemented by the Applicants' invention builds 2 occurrences of dynamic XML sub-structure 2 (from above), based on the actual transaction data (for 2 shipment items) retrieved from data sources at execution time. The complete XML per this example will look like as follows:

```
<ShipmentTransaction>
        <fromPartner>
                <Name>IBM</Name>
                <DUNS>123456789</DUNS>
        </fromPartner>
        <toPartner>
                <Name>PartnerA</Name>
                <DUNS>987654321</DUNS>
        </toPartner>
        <Shipment>
                <DocumentID>SHIP-02</ DocumentID>
                <DocumentDate>2006-01-01</ DocumentDate>
                <ShipmentType>Overnight</ShipmentType>
                <NumberOfItems>1</ NumberOfItems>
```

```
        </Shipment>
        <ShipmentItems>
                <PartNumber>PN 01/01</PartNumber>
                <PartDescription>Sample Part</PartDescription>
                <ItemNumber>0001</ItemNumber>
                <ItemQuantity>1000</ItemQuantity>
                <ExpectedDeliveryDate>2006-01-02</ExpectedDeliveryDate>
        </ShipmentItems>
        <ShipmentItems>
                <PartNumber>PN 02/01</PartNumber>
                <PartDescription>Second Sample Part</PartDescription>
                <ItemNumber>0002</ItemNumber>
                <ItemQuantity>2000</ItemQuantity>
                <ExpectedDeliveryDate>2006-01-02</ ExpectedDeliveryDate>
        </ShipmentItems>
</ShipmentTransaction>
```

Notice multiple occurrences of dynamic structure ShipmentItems

The complete XML transaction thus built will be validated against a relevant Document Type Definition (DTD) to validate the structural integrity.

According to the invention, only static sections and single-occurrence dynamic sections with empty values are pre-built, and the pre-built static and dynamic sections are linked to the TPP. Those of ordinary skill in the art would understand that a static structure is a pre-built XML structure with pre-filled values based on the associated transaction type and trading entity, and a dynamic structure includes pre-built dynamic sections with empty tags and a single occurrence of a repeating structure. The empty tags and multiple occurrences will be built at runtime.

Accordingly, the *prima facie* non-enabling rejection given in the Office Action is improper as failing to articulate the requirements for asserting such a rejection as mandated by the Federal Circuit and associated jurisdictions. Furthermore, as demonstrated above, one of ordinary skill in the art would clearly understand the Applicants' invention when reading the

specification and drawings and interpreting them in light of the prior art and their ordinary skill in the technology. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw this rejection.

III. The Prior Art Rejections

Claims 1-4, 6, 8-14, 16, and 18-21 stand rejected under 35 U.S.C. §102(b) as being anticipated by Dan, et al. (U.S. Publication No. 2002/0178103), hereinafter referred to as Dan. Claims 5, 7, 15, 17, and 22 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Dan in view of Thomas (U.S. Publication No. 2003/0167446). Applicants respectfully traverse these rejections based on the following discussion.

Dan teaches a method for automating a contract negotiation between a plurality of parties over a communications network. The parties communicate and agree upon a negotiation protocol before commencing the negotiation in a meta contract that is formed to govern or control the negotiation process. The automatic negotiation may include at least one sub negotiation. Machine-executable rules are specified to enable an automatic negotiation to take place between servers over a communications network. A successful negotiation may result in the formation of an electronic commerce contract. Each party may maintain the contract state of the overall negotiation, which may take place among two or more parties, wherein at least one party may be represented by a broker. Thus, complex negotiations may be handled automatically by the inventive method. The negotiation may be conducted semi-automatically to allow for human intervention in the negotiation process.

Thomas teaches a method of recording changes to a markup language file which employs

application-defined tags. The changes are recorded in a delta file which is also a markup language file providing validation of the recorded changes against substantially the same markup language structure as that of the markup language file being changed. Where the original markup language file is an XML file with a DTD, a DTD can be created for the delta file which substantially follows the DTD of the original markup language file. Strict compliance of the data recorded in the delta file with the delta DTD provides validation of the changes with respect to the original XML file.

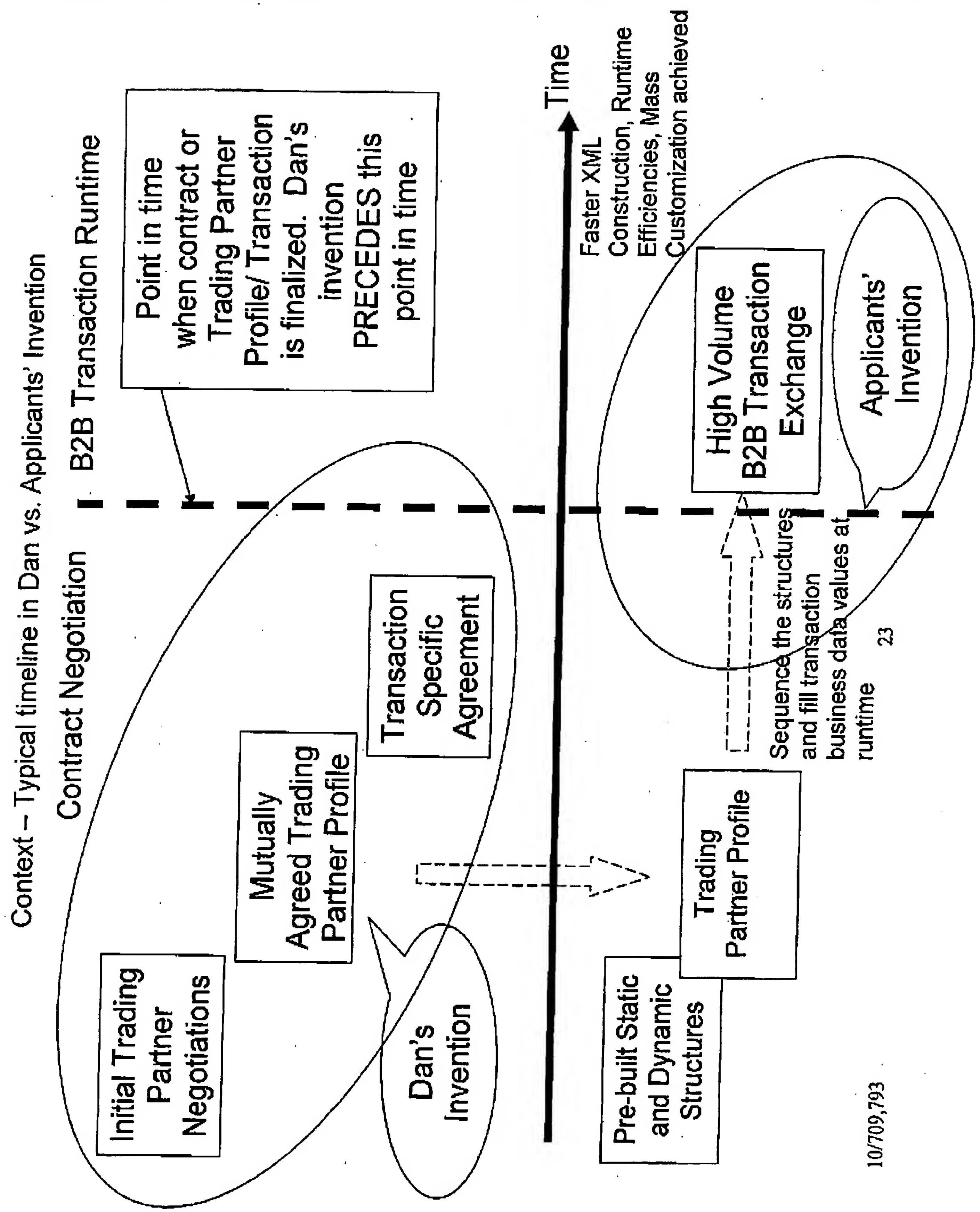
The motivation behind the Applicants' invention is that in a B2B environment, transaction information is combined using various data sources (database, files, XML format, etc.), and in a heavy B2B transaction environment, the incremental costs associated with reading and translating the stored information is significantly high. Accordingly, the Applicant's claimed invention achieves the result of speeding XML construction in high volume transaction environment by using pre-built static and dynamic structures.

Furthermore, the Applicants' claimed invention, as provided in amended independent claims 1, 11, and 21 contain features, which are patentably distinguishable from the prior art references of record. Specifically, claims 1, 11, and 21, generally provide, in part, "...<u>establishing an original pre-defined data type definition format for an XML transaction; creating a copy of said original pre-defined data type definition format for said XML transaction;</u> ... <u>wherein an occurrence of said runtime of said XML transaction occurs when said XML transaction is sent to a trading partner; and constructing a final XML structure based on the combining process, wherein said final XML structure is validated by comparing said final XML structure against said copy of said original data type definition format for said XML transaction.</u>"

These features are neither taught nor suggested in Dan, as admitted on pages 13-17 of the Office Action. Furthermore, even if Dan were properly combined with Thomas, they would still fail to teach the Applicants' claimed invention because of the occurrence of the time of occurrence of the combining of the static and dynamic structures occurs at the runtime (i.e., when said XML transaction is sent to a trading partner) of the XML transaction, whereas in Dan this occurs during the contract negotiation time (prior to the runtime of the XML transaction).

Furthermore, in Thomas the constructed XML is compared to a pre-established DTD and if there is a difference (delta) between the constructed XML and the pre-established DTD, then Thomas changes the DTD (and saves these changes into the DTD) (see Figure 3 of Thomas). Conversely, in the Applicants' claimed invention the constructed XML is compared to a copy of the pre-established DTD and if there is a difference between the constructed XML and the copy of the pre-established DTD, then the XML is invalidated (wherein said final XML structure is validated by comparing said final XML structure against said copy of said original data type definition format for said XML transaction). Therefore, in the Applicants' invention if a difference exists, then the DTD is not changed, but rather the process is repeated until no changes exist.

In other words, to use a simple analogy, if the DTD can be likened to an answer key for an exam, and the constructed XML is a student's response to an exam, then in Thomas, if there are differences between the answer key and the student's response, then the answer key is changed. However, in the Applicants' invention, if there are differences between the answer key and the student's response, then the student's response is deemed invalidated (i.e., incorrect). The differences between the Applicants' invention and Dan can be graphically shown as follows:


Context – Typical timeline in Dan vs. Applicants' Invention
Contract Negotiation
B2B Transaction Runtime
Point in time when contract or Trading Partner Profile/ Transaction is finalized. Dan's invention PRECEDES this point in time
Initial Trading Partner Negotiations
Mutually Agreed Trading Partner Profile
Transaction Specific Agreement
Dan's Invention
Time
Faster XML Construction, Runtime Efficiencies, Mass Customization achieved
High Volume B2B Transaction Exchange
Applicants' Invention
Sequence the structures and fill transaction business data values at runtime
Pre-built Static and Dynamic Structures
Trading Partner Profile

23

10/709,793

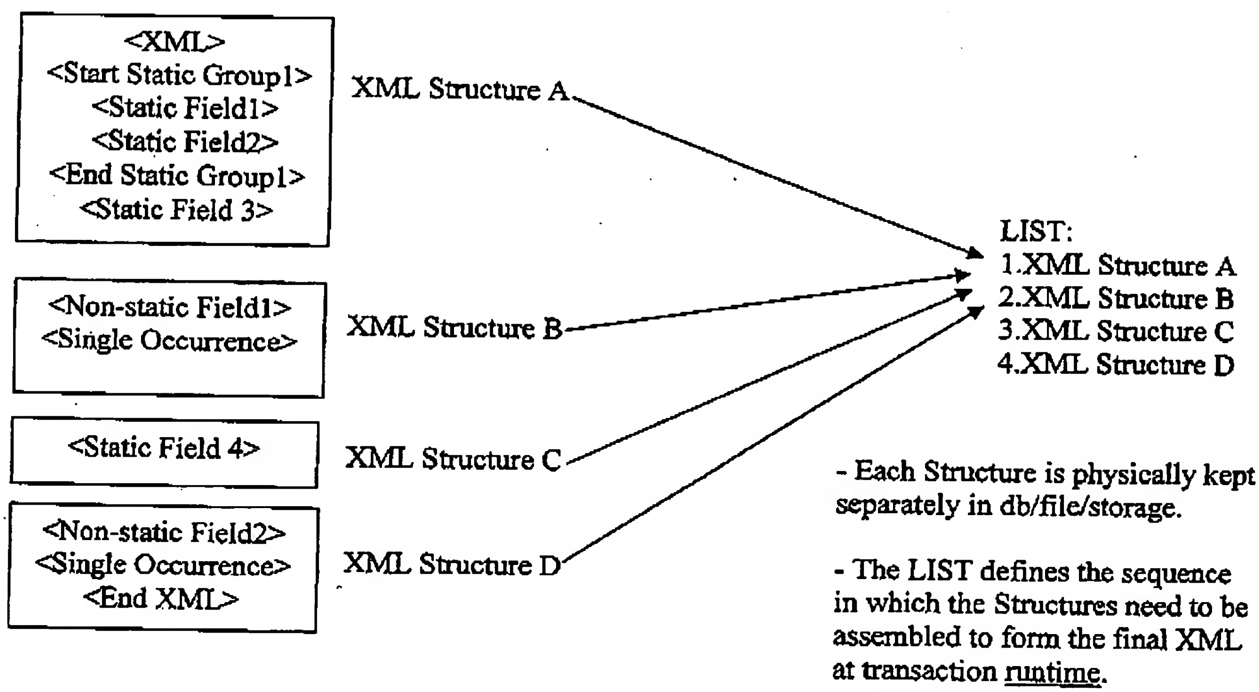
Additionally, Dan handles XML as a form template to be filled:

<XML>
<Start Static Group1>
<Static Field1>
<Static Field2>
<End Static Group1>
<Static Field 3>
<Non-static Field1>
<Static Field 4>
<Non-static Field2>
<End XML>

Values are filled in for static sections of XML

"Empty" tags are left for non-static/"negotiable" fields

Thus, Dan uses a form template with some values filled in and some values left blank. Accordingly, there is no intelligence in Dan's approach to differentiate if a field is single occurrence, multiple occurrence, etc. Conversely, the Applicants' break the XML into pieces and string them together via an external list:



The Applicants provide for pre-building of static structures of an XML transaction. The Office Action suggests that Dan teaches this as follows:

- The profile serves as the starting point of a negotiation by providing an initial version of a contract document (paragraph 30).
- The profile may be expressed, as an XML document whose contents may be incorporated into a contract (Paragraph 34).
- One example of a contract template is an almost complete electronic contract document with a few fields left blank (Paragraph 34).

However, Dan's approach leads to finalizing the structure of a contract document. Conversely, the Applicants' approach occurs after the structure of the XML is finalized between parties. Dan's approach entails filling a template with values. The template, even if it incorporates an XML, remains as a single structure. Conversely, in the Applicants' approach, the XML is broken down into fragments of several static and dynamic structures.

Next, the Applicants provide for classifying the dynamic structures of the XML transaction with empty tags and single occurrence classifiers for repeating dynamic structures. The Office Action suggests that Dan teaches this as follows:

- Negotiable field 1023 or 1024 may be treated as blank that may be completed by the negotiating party (Paragraph 35).

However, Dan shows the method of leaving few fields blank in a document template, while it is being exchanged between partners. In others words, the document is transmitted with blank values. Conversely, the Applicants' approach does not involve leaving fields blank in a transaction when exchanged between partners. Additionally, Dan talks only about blank fields.

Whereas, the Applicants provide a mechanism where an entire sub-structure within an XML can be left with empty tags with attributes embedded in the XML itself or as an attribute of the list indicating if the entire structure is a single or repeating occurrence. Furthermore, as indicated above, Dan's approach has no intelligence on whether the "blank" field is to be filled. Conversely, the Applicants mark the pieces of XML that are truly dynamic and provide a method for empty tags, and classifiers to indicate if the structure is single occurrence, multiple occurrences, etc.

Next, the Applicants provide for building a list of a sequence of static and dynamic structures. The Office Action suggests that Dan teaches this as follows:

- A set of sequencing rules 180 may be provided in meta contract 110 to ensure that the various negotiation actions are being issued in the correct order (Paragraph 32).

However, the Applicants list is different from that taught in Dan as described above. Moreover, Dan's use of sequencing rules is fundamentally different than the Applicants' approach. Dan's sequencing rules refer to the *choreography* of a series of <u>negotiation actions</u> as part of a <u>long-running</u> transaction. Dan himself defines the various <u>Action Names</u> in his Paragraph 42. This clearly demonstrates that Dan does not teach the Applicants' claimed invention.

Next, the Applicants provide for linking the list to a type of XML transaction and a predetermined trading partner profile. The Office Action suggests that Dan teaches this as follows:

– Starting definitions and values for these types of information in the negotiated contract may be provided in a TPA template or party profile (Paragraph 32).

Dan provides starting definitions and values from a TPA template or party profile. For example, Dan might derive starting values from the TPA template for Company ABC. However, the Applicants' approach links the list of static and dynamic parts to the defined Trading Partner Profile and specific transaction type. For example, the Applicants could link the list to Company ABC and a shipping transaction type. Dan does not and cannot accomplish this.

Furthermore, the Applicants provide for combining the static structures with the dynamic structures at a runtime of the XML transaction based on the sequence, the type of XML transaction, the partner profile, and the dynamic structures of XML transaction. The Office Action suggests that Dan teaches this as follows:

– One example of a contract template is an almost complete electronic contract document with a few fields left blank (Paragraph 34).
– Once a contract template of Dan is sent for negotiation, it contains fields that are set and non-negotiable, and fields that are negotiable.

However, Dan teaches how to fill a form with some fields filled in and some fields left blank. Dan's teaching transmits a template with blank and non-blank fields. Moreover, Dan does not teach how to break an XML transaction, and classify them as static and dynamic components. Furthermore, Dan does not teach how to dynamically fill values in an XML to construct a complete XML prior to transmission. Rather, Dan teaches how to send blank tags to be optionally filled by receiving partners. Moreover, Dan's teaching does not assemble the complete XML (with no blank field/structures) prior to transmission. Dan's teaching involves

two or more partners to fill the blank fields. Conversely, the Applicants' approach involves breaking and assembling the XML within a single partner's environment before transmitting the information to another partner. Moreover, the Applicants combine static and dynamic structures at runtime, with actual transaction values, at runtime, when the complete transaction is sent to the partner. Additionally, Dan's method talks about building an almost complete electronic contract document prior to runtime. In the Applicants' approach, actual transaction values can differ for each individual transaction (ex. shipped product, quantity and dollar value in today's transaction can be different from the very next instance of the transaction for the same partner, and same transaction type).

Furthermore, the Applicants provide for pre-building of the static structures to occur prior to runtime of the XML transaction. The Office Action suggests that Dan teaches this as follows:

- The profile serves as the starting point of a negotiation by providing an initial version of a contract document (Paragraph 33).
- The profile may be expressed, as an XML document whose contents may be incorporated into a contract (Paragraph 34).
- One example of a contract template is an almost complete electronic contract document with a few fields left blank (Paragraph 34).
- Contract of Dan runs once the negotiation phase begins to fill in the initial blank negotiable fields 1023 and 1024.

However, as mentioned Dan teaches how to fill a form with some fields filled in and transmit a template with blank and non-blank fields. But, Dan does not teach how to break an XML transaction and classify the static components. Dan teaches how to fill fields in a template

as part of the process of building and transmitting an instance of the XML. This <u>entire</u> process occurs once per document transmission. Conversely, the Applicants' approach involves two distinct events. In a first phase, the static component is classified and filled in. The phase first occurs one time for a partner and transaction type. The second phase occurs every time a transaction is sent to a partner, wherein the static structures are combined with the dynamic structures, according to the sequence defined by the list.

Next, the Applicants provide for filling the empty tags of the dynamic structures with business data values. The Office Action suggests that Dan teaches this as follows:

- Negotiable 1023 or 1024 may be treated as a blank that may be completed by the negotiating party (Paragraph 35).

However, again, Dan teaches how to fill a form with some fields filled in and transmit a template with blank and non-blank fields. Dan does not teach how to break an XML transaction and classify the static and dynamic components. Rather, Dan teaches how to fill fields in a template as part of the process of building and transmitting an instance of the XML. Again, this <u>entire</u> process occurs once per document transmission. Conversely, the Applicants' approach involves two distinct steps. In the first phase, the static component is classified and filled in. The first phase occurs one time for a partner <u>and</u> transaction type. The second phase occurs every time a transaction is sent to a partner, wherein the dynamic structures are automatically <u>filled</u> based on the associated pre-defined business logic, and the static structures are combined with the dynamic structures according to the sequence defined by the list. Moreover, the Applicants' approach provides the manner of filling actual business data values in the dynamic sections of an XML to construct a complete XML prior to transmission. Whereas, Dan teaches

how to send blank tags to be optionally filled by receiving partners. Furthermore, the Applicants' approach provides for expanding dynamic structures (ex. multiple occurrences) at runtime, based on actual transaction business data values occurring at runtime.

Next, the Applicants provide for creating a copy of a pre-defined data type definition format comprising the XML format. The Office Action suggests that Thomas teaches this as follows:

- The processor reads 12 the document type definition (DTD) of the first XML file and creates copy 13 of the DTD.

However, the Applicants use the DTD to validate the final XML after combining static and dynamic structures, and filling actual transaction business data values at runtime. As previously mentioned, the manner of comparison and validation is different between Thomas the Applicants' claimed invention.

In view of the foregoing, the Applicant respectfully submits that the collective cited prior art do not teach or suggest the features defined by amended independent claims 1, 11, and 21 and as such, claims 1, 11, and 21 are patentable over Dan alone or in combination with Thomas. Further, dependent claims 2-6, 8-10, 12-16, 18-20, and 22-24 are similarly patentable over Dan alone or in combination with Thomas, not only by virtue of their dependency from patentable independent claims, respectively, but also by virtue of the additional features of the invention they define. Thus, the Applicant respectfully requests that these rejections be reconsidered and withdrawn. Moreover, the Applicant notes that all claims are properly supported in the specification and accompanying drawings. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections.

RECEIVED
CENTRAL FAX CENTER
NOV 1 6 2006

IV. Formal Matters and Conclusion

With respect to the rejections to the drawings, the specification has s been amended, to overcome these rejections. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections to the drawings. With respect to the rejections to the claims, the claims have been amended, above, to overcome these rejections. In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw the rejections to the claims.

In view of the foregoing, Applicants submit that claims 1-6, 8-16, and 18-24, all the claims presently pending in the application, are patentably distinct from the prior art of record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary. Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0456.

Respectfully submitted,

Dated: November 16, 2006

Mohammad S. Rahman
Registration No. 43,029

Gibb I.P. Law Firm, LLC
2568-A Riva Road, Suite 304
Annapolis, MD 21401
Voice: (301) 261-8625
Fax: (301) 261-8825
Customer Number: 29154